CELEBAL TECHNOLOGIES PVT. LTD.

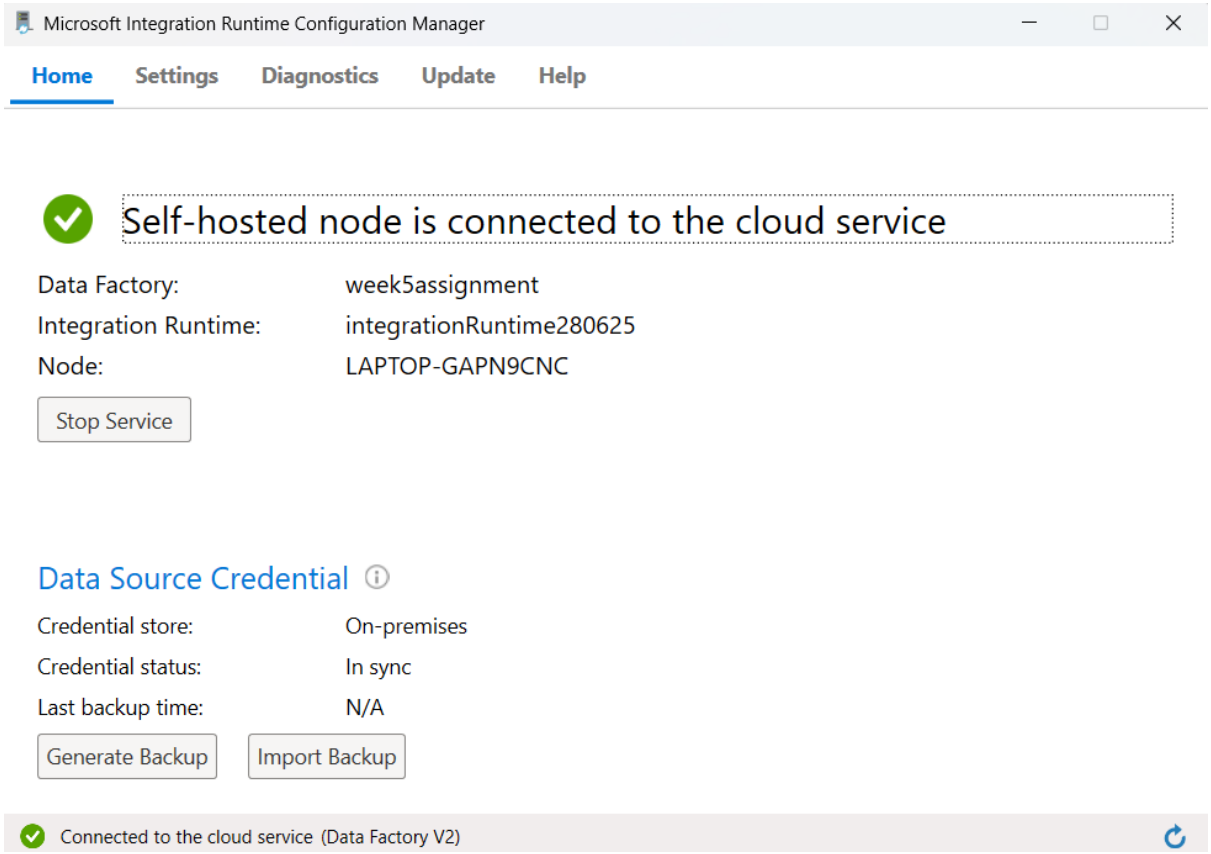
ASSIGNMENT

WEEK 6

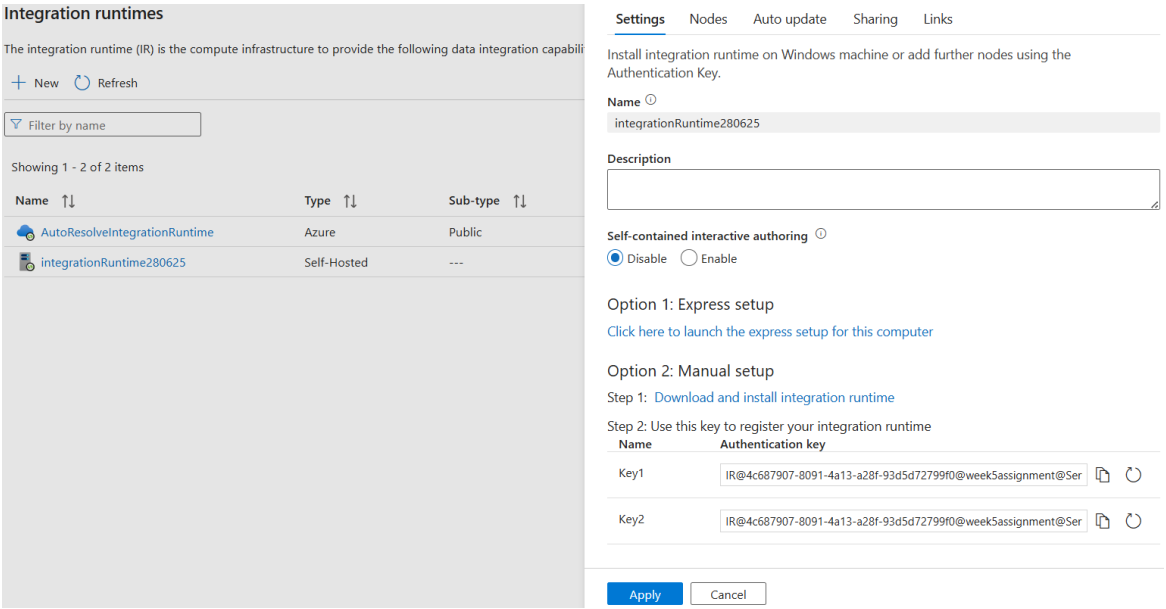
Configure Self-hosted Integration Runtime to Extract Data from Local Server and Load into Azure SQL Database: Set up a Self-hosted Integration Runtime (SHIR) in Azure Data Factory to securely connect to an on-premises server. Extract data from the local environment and load it into an Azure SQL Database to enable cloud-based analytics and processing.

STEP 1: Download and install Microsoft Integration Runtime Manager.

STEP 2: Link Azure Data Factory



STEP 3: Create new Integration Runtime(integrationRuntime280625) in Azure Data Factory



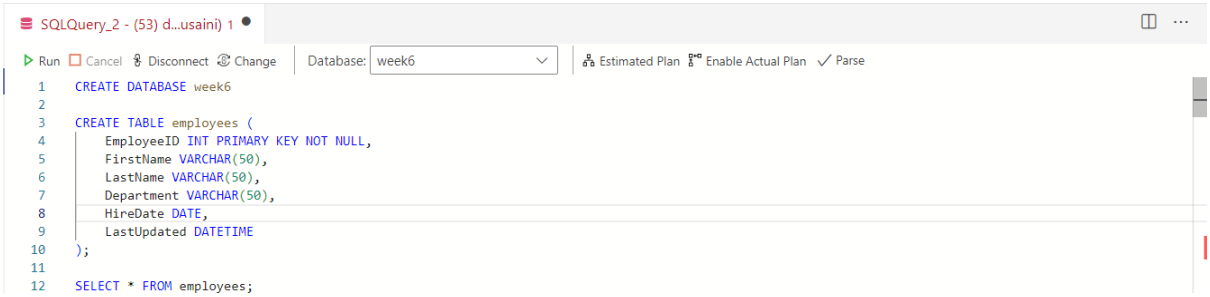
Tested Using: Copy data activity(localDb2-azureSQLDatabase)

DB2 (Source) Queries:

```
db2 => connect to week6

   Database Connection Information

 Database server        = DB2/NT64 12.1.1.0
 SQL authorization ID   = DEEPANSH...
 Local database alias   = WEEK6
```

```
db2 => CREATE TABLE employees (EmployeeID INT PRIMARY KEY NOT NULL, FirstName VARCHAR(50), LastName VARCHAR(50), Department VARCHAR(50), HireDate DATE, Last
Updated TIMESTAMP);
DB20000I  The SQL command completed successfully.
db2 => INSERT INTO Employees (EmployeeID, FirstName, LastName, Department, HireDate, LastUpdated) VALUES (1, 'DEEP', 'KAUR', 'HR', '2023-01-15', CURRENT_TIM
ESTAMP), (2, 'ARYA', 'SHARMA', 'IT', '2022-12-05', CURRENT_TIMESTAMP), (3, 'RIYA', 'SINGH', 'Finance', '2021-06-20', CURRENT_TIMESTAMP), (4, 'ASHA', 'SAINI'
, 'Marketing', '2020-11-11', CURRENT_TIMESTAMP);
DB20000I  The SQL command completed successfully.
db2 => SELECT * FROM Employees FETCH FIRST 1 ROWS ONLY;

EMPLOYEEID  FIRSTNAME                                         LASTNAME                                          DEPARTMENT
          HIREDATE    LASTUPDATED
----------- ------------------------------------------------- ------------------------------------------------- ----------------------------------------
---------- ---------- --------------------------
          1 DEEP                                              KAUR                                              HR
         01/15/2023 2025-06-28-16.19.43.076000

  1 record(s) selected.
```

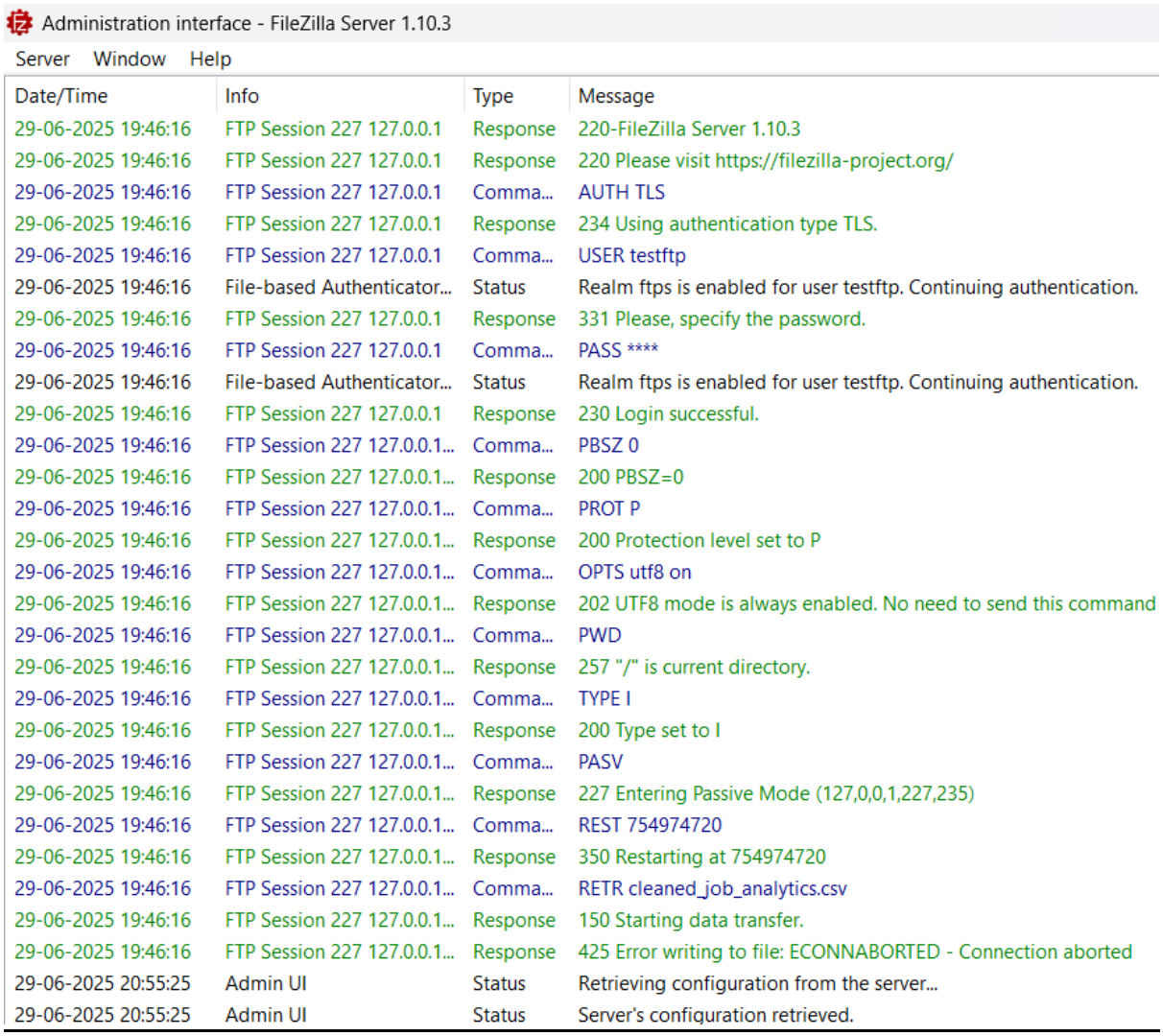
Azure Data Studio (Sink) Queries:



Configure FTP/SFTP Server and Create an ADF Pipeline for Data Extraction:

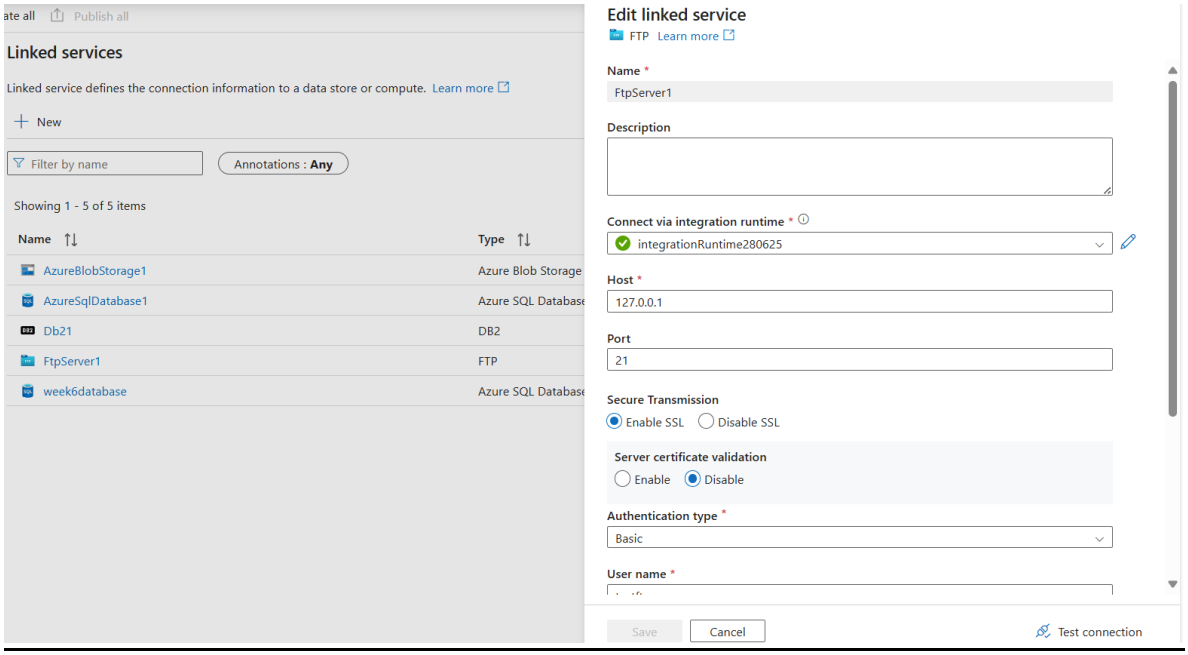
Set up access to FTP/SFTP servers and build a pipeline in Azure Data Factory to extract files or datasets from these sources. This facilitates integration with external systems or partners.

STEP 1: Log in to FTP server.

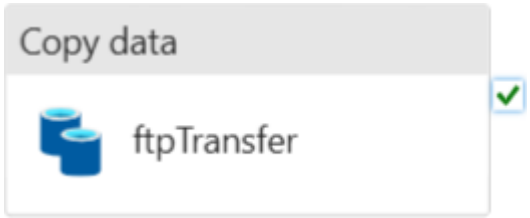


Date/Time	Info	Type	Message
29-06-2025 19:46:16	FTP Session 227 127.0.0.1	Response	220-FileZilla Server 1.10.3
29-06-2025 19:46:16	FTP Session 227 127.0.0.1	Response	220 Please visit https://filezilla-project.org/
29-06-2025 19:46:16	FTP Session 227 127.0.0.1	Comma...	AUTH TLS
29-06-2025 19:46:16	FTP Session 227 127.0.0.1	Response	234 Using authentication type TLS.
29-06-2025 19:46:16	FTP Session 227 127.0.0.1	Comma...	USER testftp
29-06-2025 19:46:16	File-based Authenticator...	Status	Realm ftps is enabled for user testftp. Continuing authentication.
29-06-2025 19:46:16	FTP Session 227 127.0.0.1	Response	331 Please, specify the password.
29-06-2025 19:46:16	FTP Session 227 127.0.0.1	Comma...	PASS ****
29-06-2025 19:46:16	File-based Authenticator...	Status	Realm ftps is enabled for user testftp. Continuing authentication.
29-06-2025 19:46:16	FTP Session 227 127.0.0.1	Response	230 Login successful.
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Comma...	PBSZ 0
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Response	200 PBSZ=0
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Comma...	PROT P
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Response	200 Protection level set to P
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Comma...	OPTS utf8 on
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Response	202 UTF8 mode is always enabled. No need to send this command
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Comma...	PWD
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Response	257 "/" is current directory.
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Comma...	TYPE I
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Response	200 Type set to I
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Comma...	PASV
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Response	227 Entering Passive Mode (127,0,0,1,227,235)
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Comma...	REST 754974720
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Response	350 Restarting at 754974720
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Comma...	RETR cleaned_job_analytics.csv
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Response	150 Starting data transfer.
29-06-2025 19:46:16	FTP Session 227 127.0.0.1...	Response	425 Error writing to file: ECONNABORTED - Connection aborted
29-06-2025 20:55:25	Admin UI	Status	Retrieving configuration from the server...
29-06-2025 20:55:25	Admin UI	Status	Server's configuration retrieved.

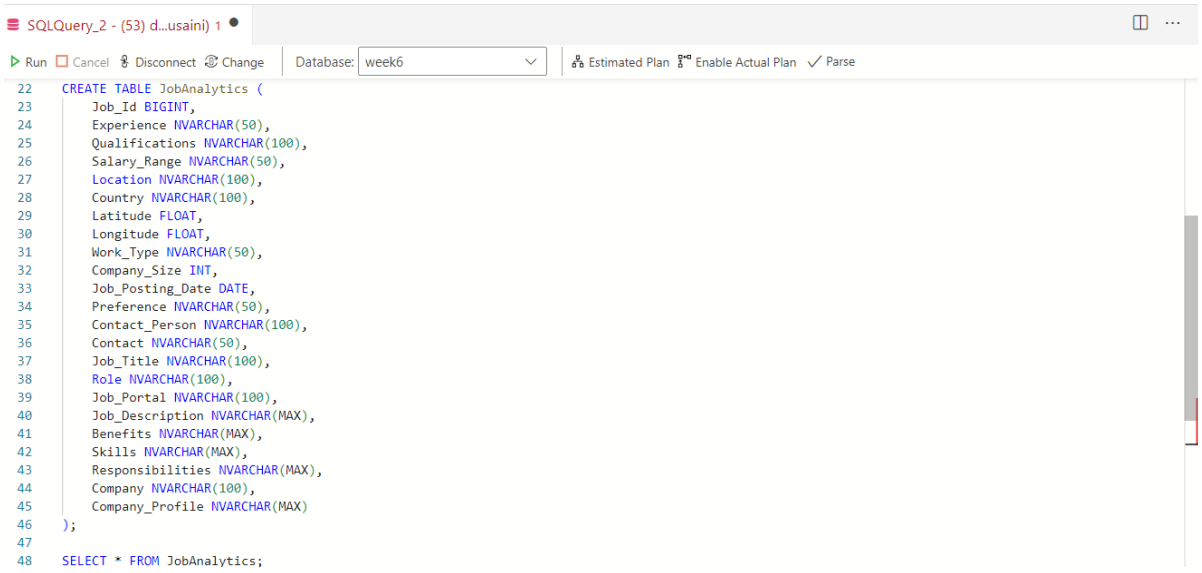
STEP 2: Create FTP Linked Service (FtpServer1) on Azure Data Factory



Tested Using: Copy data Activity (ftpTransfer)

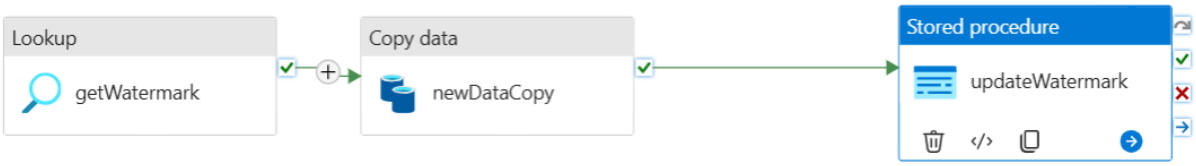


Azure Data Studio (Sink) Queries:

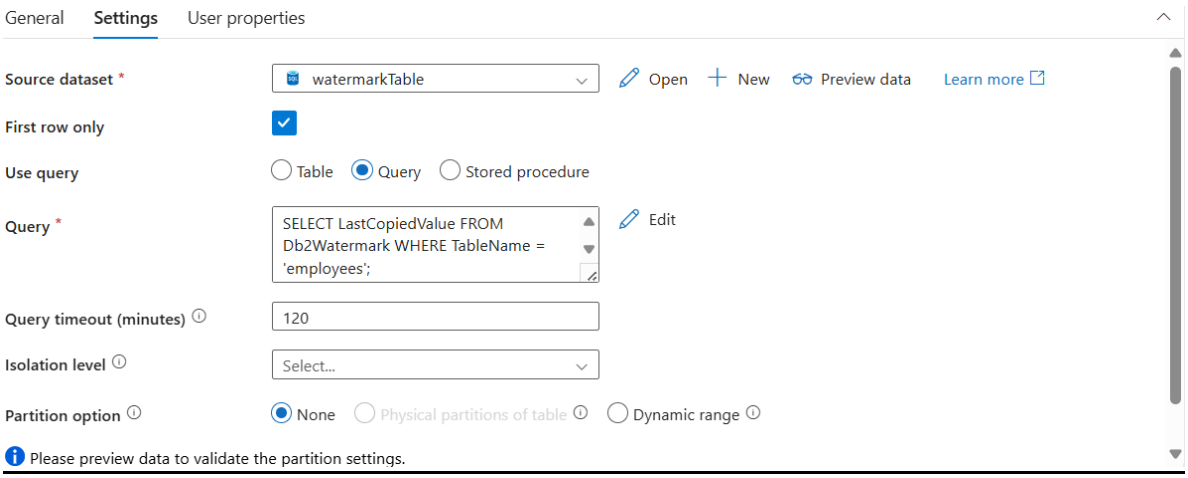


Create Incremental Load Pipeline and Automate This on a Daily Basis: Design an ADF pipeline to perform incremental data loads using techniques like watermarking or change tracking. Schedule this pipeline to run daily to ensure

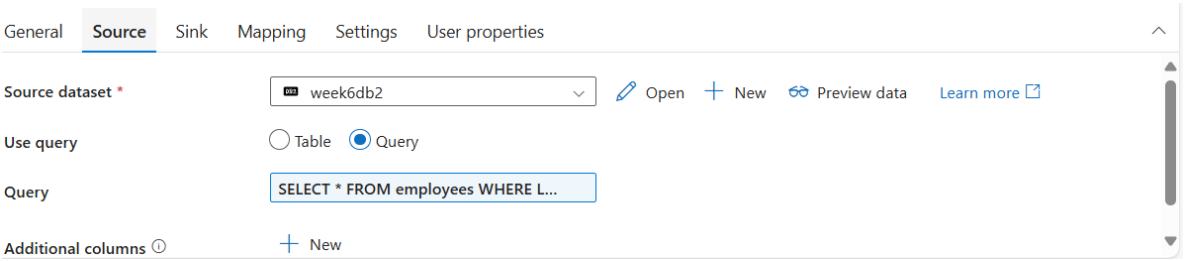
only new or modified records are processed, optimizing performance and reducing load.

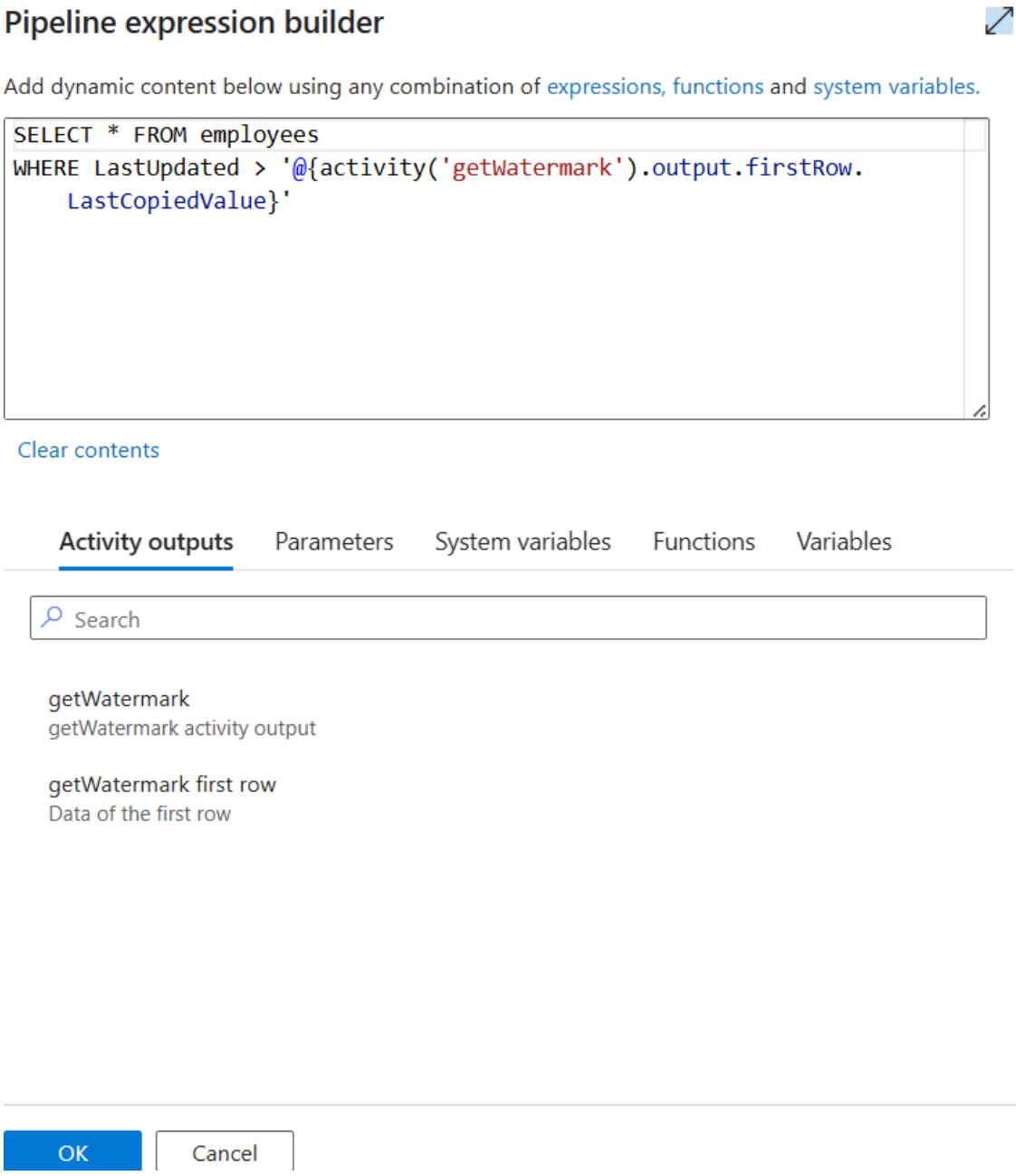


STEP 1: Lookup Activity (getWatermark)

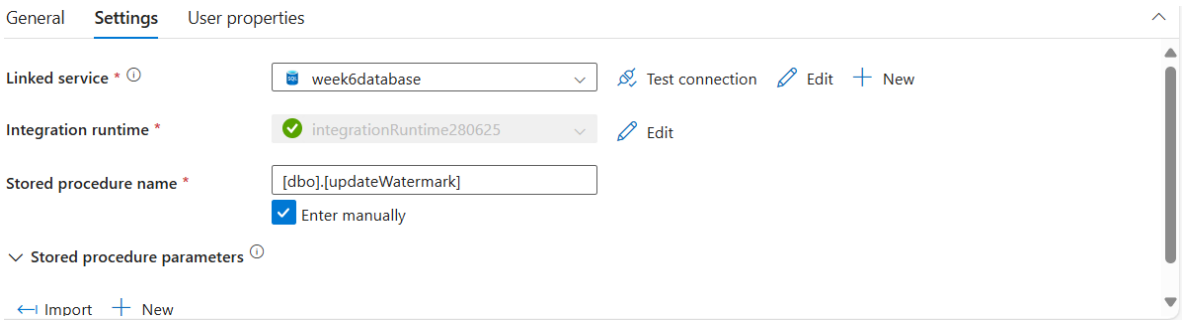


STEP 2: Copy data Activity (newDataCopy)

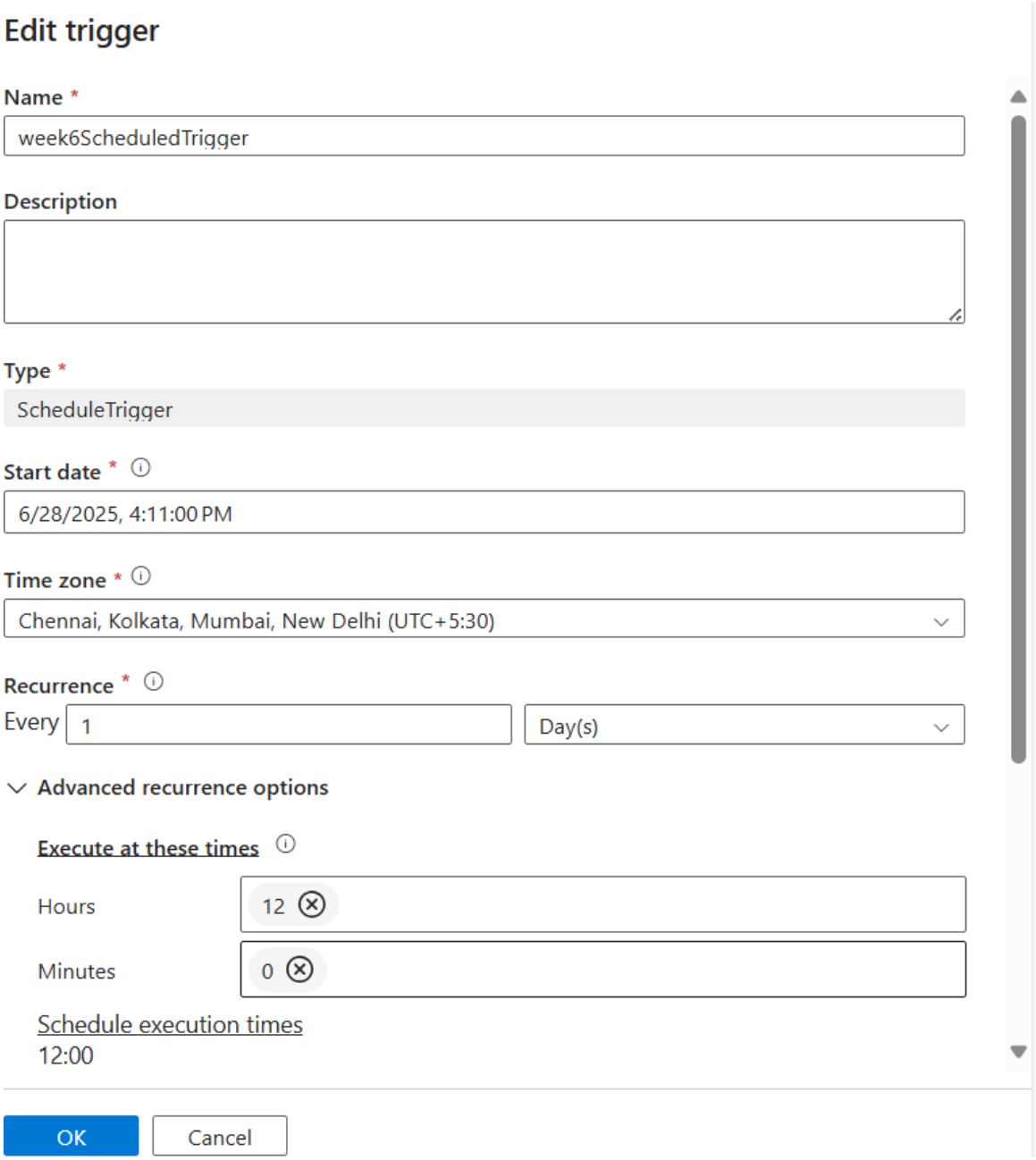




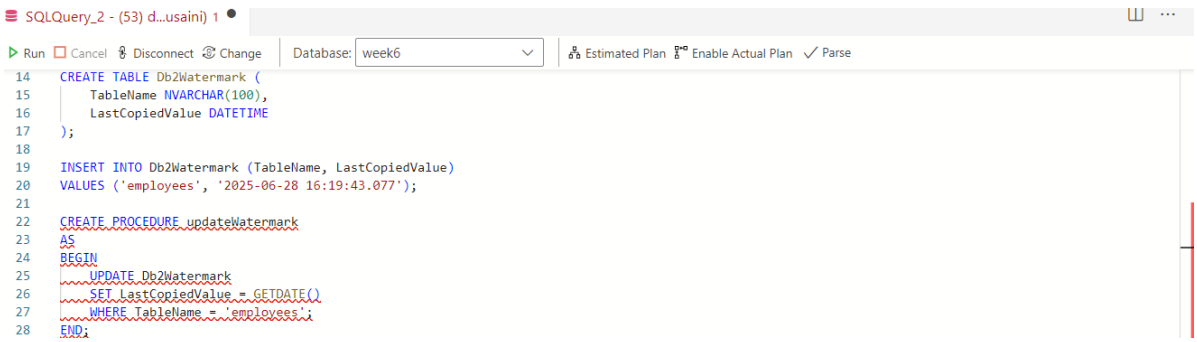
STEP 3: Stored procedure Activity (updateWatermark)



STEP 4: Scheduled Trigger (week6ScheduledTrigger)



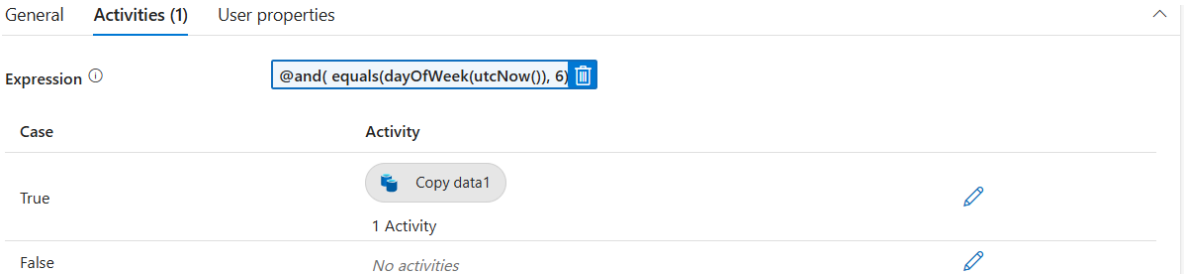
Azure Data Studio Queries:

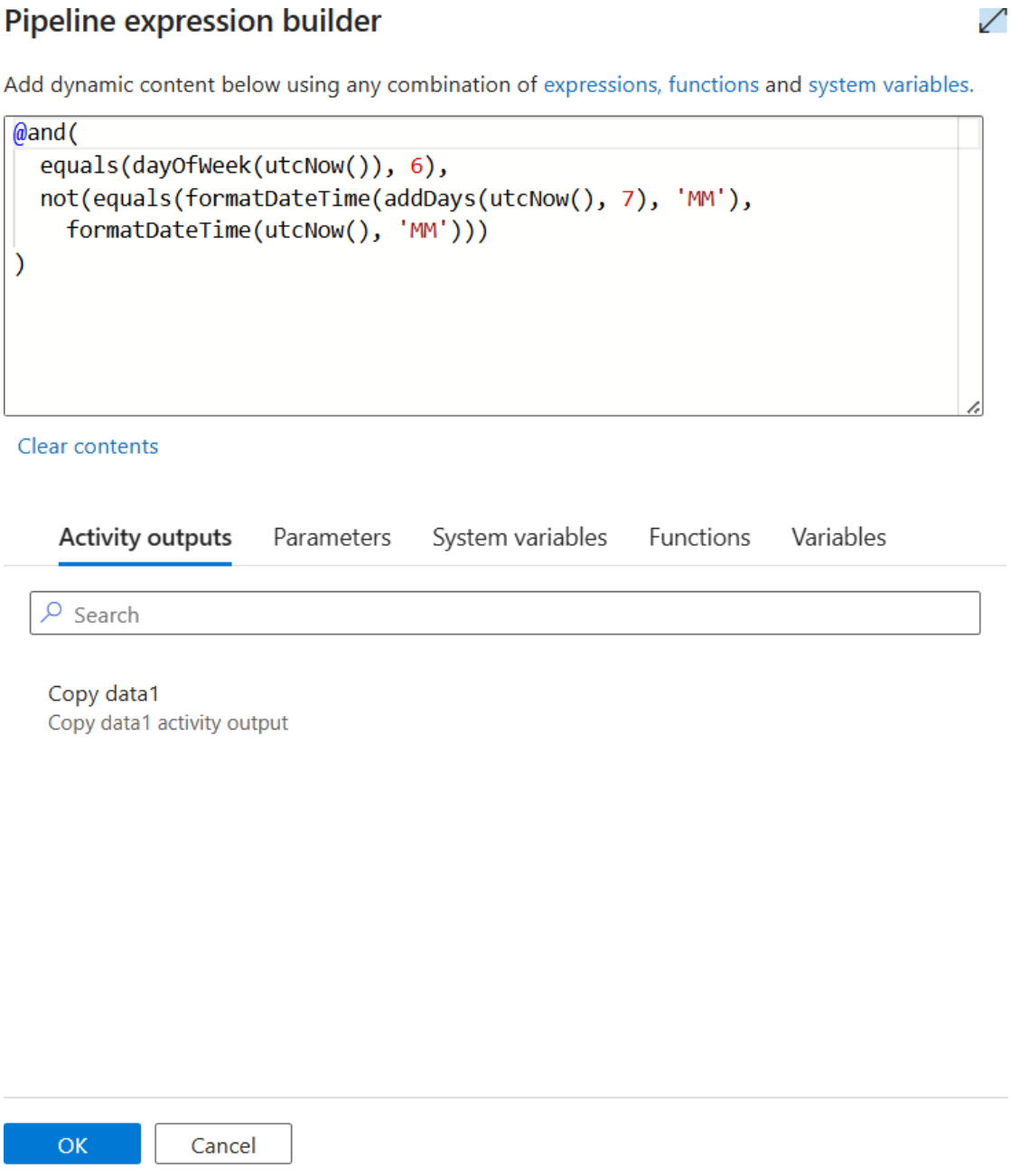


Automate a Pipeline to Trigger Every Last Saturday of the Month: Configure a custom time-based trigger in Azure Data Factory to run the pipeline on the last

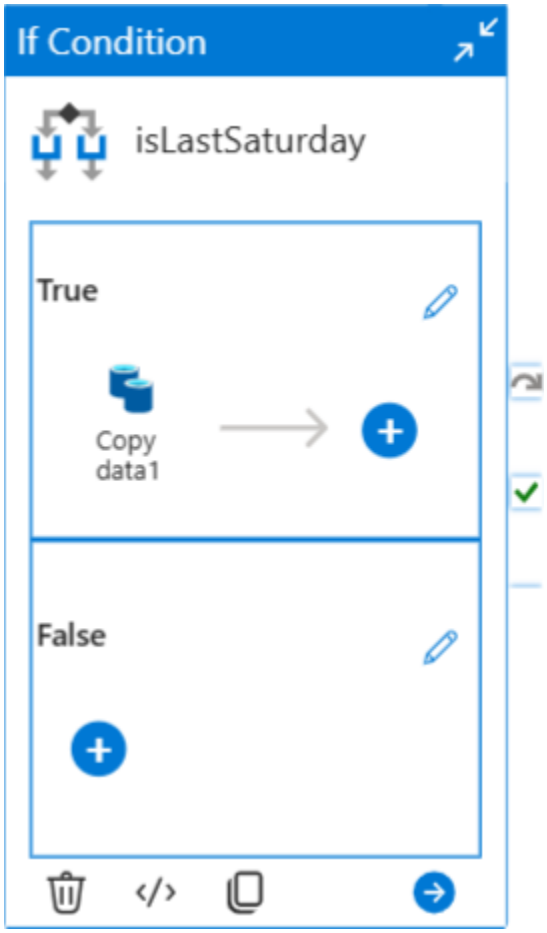
Saturday of each month. This supports periodic reporting and batch processing without manual intervention.

STEP 1: If Condition Activity (isLastSaturday)





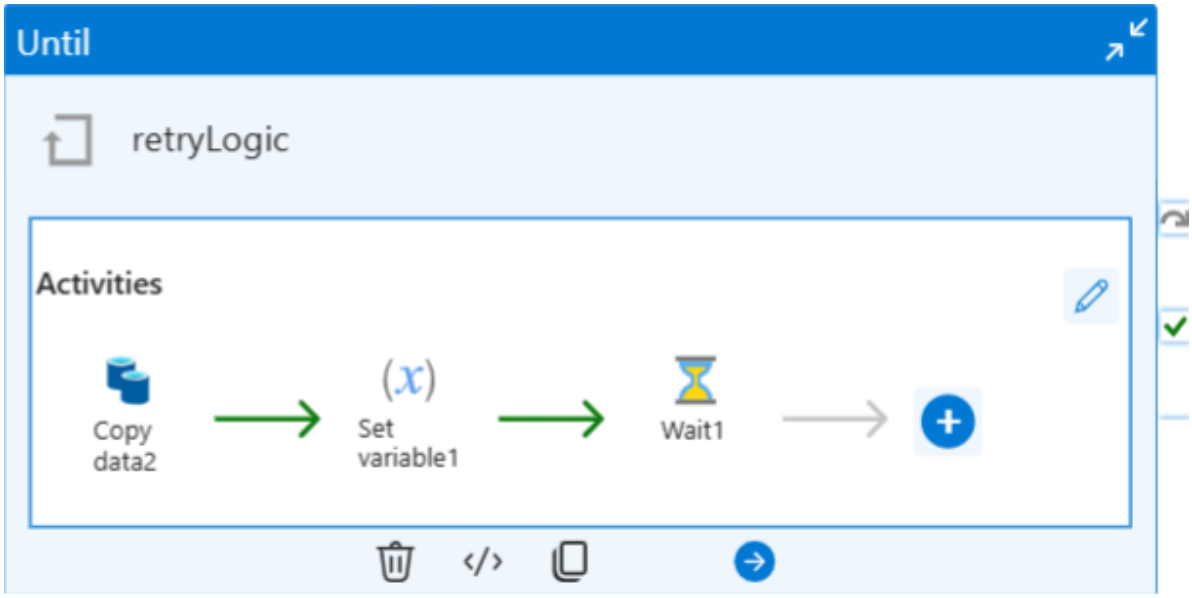
- Copy data Activity (Copy data1) inside True Case



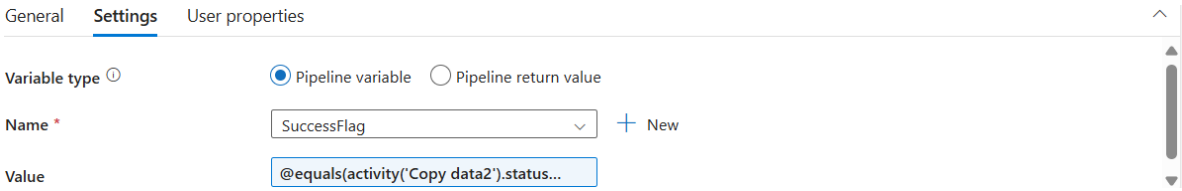
Retrieving data. Wait a few seconds and try to cut or copy again: Implement logic to handle data retrieval failures gracefully by waiting briefly (e.g., a few seconds) before retrying cut, copy, or extraction operations, improving pipeline resilience and reducing transient error impacts.

STEP 1: Until Activity (retryLogic)

- Copy data Activity (Copy data2)



- Set variable Activity (Set variable1)



Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variables.

```
@equals(activity('Copy data2').status, 'Succeeded')
```

Clear contents

Activity outputs | Parameters | System variables | Functions | Variables

Search

Copy data2
Copy data2 activity output

OK | Cancel

- Wait Activity (Wait1)

General | Settings | User properties

Wait time in seconds * 5